

# ME R-CNN: Multi-Expert Region-based CNN for Object Detection

Hyungtae Lee<sup>†‡</sup>, Sungmin Eum<sup>†§</sup>, Heesung Kwon<sup>†</sup>

<sup>†</sup>U.S. Army Research Laboratory, Adelphi, MD, USA

<sup>‡</sup>Booz Allen Hamilton Inc., McLean, VA, USA

<sup>§</sup>Institute of Advanced Computer Studies, University of Maryland, College Park, MD, USA

lee.hyungtae@bah.com   smeum@umiacs.umd.edu   heesung.kwon.civ@mail.mil

## Abstract

*Recent CNN-based object detection methods have drastically improved their performances but still use a single classifier as opposed to "multiple experts" in categorizing objects. The main motivation of introducing multi-experts is twofold: i) to allow different experts to specialize in different fundamental object shape priors and ii) to better capture the appearance variations caused by different poses and viewing angles. The proposed approach, referred to as multi-expert Region-based CNN (ME R-CNN), consists of three experts each responsible for objects with particular shapes: horizontally elongated, square-like, and vertically elongated. Each expert is a network with multiple fully connected layers and all the experts are preceded by a shared network which consists of multiple convolutional layers.*

*On top of using selective search which provides a compact, yet effective set of region of interests (RoIs) for object detection, we augmented the set by also employing the exhaustive search for training. Incorporating the exhaustive search can provide complementary advantages: i) it captures the multitude of neighboring RoIs missed by the selective search, and thus ii) provide significantly larger amount of training examples to achieve the enhanced accuracy.*

use multiple experts, each associated with the only corresponding shape patterns or priors, in order to better capture variations of object appearance. [11, 21, 27]

Although recently introduced convolutional neural network (CNN) has shown notable performance in object detection, none of those methods adopted multiple experts. The state-of-the-art CNN-based object detection methods such as the region-based CNN (R-CNN) [14] and its descendants [7, 13, 16, 28] inherit the single stream architecture of the image classification CNN which consists of sequentially connected layers [17, 23, 31, 32]. This kind of single stream architecture inheritance was found to be natural, effective, and convenient because: 1) they wanted to harvest the state-of-the-art performance of the image classification CNN pre-trained on an extremely large-scale ImageNet dataset [9] and 2) currently available datasets designed for object detection do not contain sufficient number of images and label information to train a high performance network from scratch.

We introduce a novel CNN-based approach for object detection, referred to as ME R-CNN, which uses multiple expert classifiers in place of a conventional single classifier incorporated in CNN-based object detection architectures. In this paper, ME R-CNN is built on recent variations of R-CNN, such as Fast R-CNN (or Faster R-CNN) by substituting the single classifier for multi-experts. It is important to note that the concept of multi-expert is not limited to the R-CNN based approaches but can be easily extended into any CNN-based object detection architecture using a single classifier.

Within the ME R-CNN architecture, the regions-of-interest (RoIs) are first categorized into three fundamental object shape categories according to their aspect ratios: horizontally elongated, square-like, and vertically elongated. Then each RoI is processed by the appropriate expert which specializes in handling the corresponding shape category. Each expert is constructed by connecting several fully connected layers, and all the experts are preceded by a single RoI pooling layer and a set of shared convolutional layers.

## 1. Introduction

In general, object detection uses distinctive shape patterns as evidence to find the object-of-interest in an image. Object detection models are trained on these shape patterns that are commonly shown within the same object categories yet discriminative among the different categories. However, it is burdensome for a single model to accurately identify all the appearances since object appearances greatly vary according to fundamental object shape priors (e.g., airplane vs. person) as well as different object poses and viewing angles (e.g., a person lying down vs. standing upright). Therefore, several conventional object detection methods

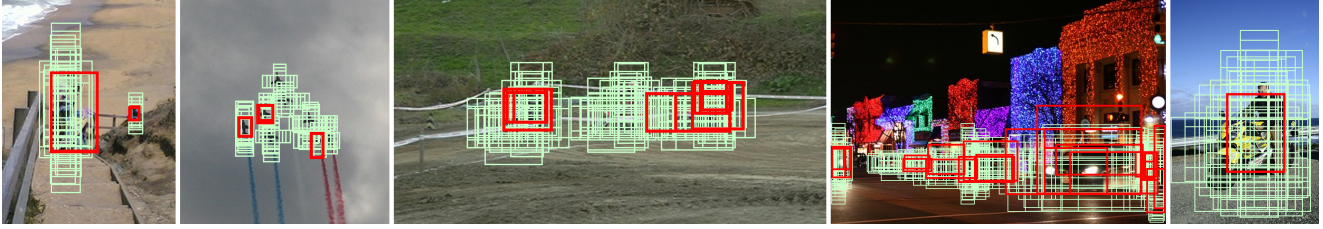


Figure 1: **Complementary roles of the selective and exhaustive search.** RoIs generated by the selective and the exhaustive search for five sample images are indicated by red and green bounding boxes, respectively. While the selective search provides compact number of RoIs with better localization, some objects can be missed (e.g., aeroplanes in the second image). The exhaustive search provides much larger number of RoIs rarely missing the target object regions.

The weights in all the experts are initialized by those of the fully connected layers in the image classification CNN. Each expert is fine-tuned separately only by using the RoIs with the associated shape category.

The major change brought into ME R-CNN compared to the single-stream image classification CNN is that the network has expanded in width, where the network width refers to the number of nodes in each layer. This follows the philosophy that learning the wider CNN can provide better discrimination power. There have already been several attempts where the width of CNN was expanded. Krizhevsky et al. [23] splits each layer into two parallel layers in order to fully use two GPUs in a parallel fashion. Szegedy et al. [32] uses the inception module which widens the network by employing multiple parallel layers using filters of different sizes. As these previous approaches proposed new architectures with “wider width” in their own ways, the networks had to be learned from scratch. However, our method aims to provide a more practical way to expand the width of the network by fully harvesting the power of the pretrained networks and guarantee a leap in performance.

In this paper, we also focused on augmenting the training data for learning ME R-CNN. It is a notion widely agreed upon, that more training data serves as a better source to learn a model with an enhanced accuracy. One way is to augment the training data by combining multiple datasets, for instance, using both PASCAL VOC and Microsoft COCO for training. Instead of bringing more examples from more datasets, we use a method to acquire more training examples by generating more RoIs for each image, equivalent to providing more training examples within the same dataset.

Recently introduced R-CNNs [7, 13, 14, 16, 28], which we share the structure with, use approximately 2000 RoIs per image generated by selective search. For ME R-CNN, we have employed a multi-scale sliding window searching strategy (exhaustive search) along with the selective search in order to acquire an augmented set of RoIs. The augmented set of RoIs provided by the exhaustive search has its own complementary roles in training ME R-CNN as fol-

lows.

1. The exhaustive search can localize a multitude of closely neighboring regions of objects which carries valuable visual contents for training, most of which are missed by the selective search.
2. An incomparably large number of regions can be acquired as positive examples as well as negative examples, providing rich source of information to train the CNN.

Figure 1 shows these complementary roles of the exhaustive search. Note that, for testing, we only use a sparse set of RoIs generated by the selective search to maintain the efficiency of the testing process.

To show that the proposed architecture can be effectively implanted into various types of object detection CNN architectures, we have built ME R-CNN on top of two widely used object detection CNN architectures which are Fast R-CNN and Faster R-CNN. For both cases, we verified that ME R-CNN can provide constant performance boost over the baseline approaches in both PASCAL VOC 07 and 12 datasets.

The contributions of the proposed ME R-CNN can be summarized as follows.

1. We adopt “multiple experts” which can be used as a module to be put into any object detection CNN architectures: each expert only focusing on classifying object instances with similar shape categories based on the form of aspect ratios.
2. The exhaustive search is complementary to the selective search, and thus we train the network on RoIs generated by both searching approaches.
3. ME R-CNN provides a significant performance boost over the baseline approaches (i.e., CNNs using neither multi-experts nor exhaustive search) on two benchmark datasets: PASCAL VOC 07 and 12.

## 2. Related Works

Object detection is one of the most challenging tasks in computer vision. Prior to the introduction of CNNs, non-CNN based machine learning approaches, such as SVM, DPM, etc., were widely used for classifying RoIs into corresponding object categories [8, 11, 21, 27] to produce detection results. Within the past decade, several successful attempts have been made to use CNNs for object detection. Prominent methods among them are R-CNN [14] and its descendants [7, 13, 16, 28] that provided the state-of-the-art performance for both localization accuracy and speed. Although having achieved the top-notch performance, R-CNNs do not exploit some of the strategies which conventional object detection methods commonly use for boosting the performance.

The first attempt to add a common performance boosting strategy to the R-CNNs is done by Shrivastava et al. [30]. Instead of using the heuristic hard negative examples of Fast R-CNN, they used the online hard example mining (OHEM) to automatically select hard examples with high optimization loss in every iteration of training. Motivated by their successful practice, two conventional performance boosting strategies have been used in our method which enhance the network in two different aspects: i) introducing multi-experts associated with shape categories and ii) using a complementary combination of the exhaustive and selective search. We incorporate the two components into the Fast R-CNN architecture to construct ME R-CNN.

**Mixture-of-Experts Models.** Multiple experts embedded in the proposed ME R-CNN is based on the concept of mixture-of-experts models. The mixture-of-experts model is used to better estimate the probability distribution of a composite data with large variation (e.g., Gaussian mixture model [36]). In the image domain, object appearances can also show large variations according to their poses and viewing angles. Figure 2 in Felzenswalb et al. [11] nicely illustrates the importance of using a mixture of models by presenting two models, each of which captures the appearance of the front and the side view of a bicycle. Accordingly, many recent approaches [3, 11, 29] have shown that using the mixture-of-experts model for advanced object detection is very effective. However, to date to the best of our knowledge, none of the CNN-based object detection methods have incorporated the mixture-of-experts model into their architectures. ME R-CNN proposes a novel architecture, which consists of a shared set of convolutional layers followed by multiple sets of fully connected networks, referred to as ‘multi-expert networks.’

**RoI Generation.** One of the conventional ways to generate RoIs is to use multi-scale sliding windows [2, 8, 11, 15, 21, 27, 35] which can be considered as a ‘dense’ search.

To avoid impractical computational complexity, the search space is confined to a regular grid and a fixed set of scales and aspect ratios. The branch and bound strategy was found to reduce the search space even more by using optimal windows within an image [22, 34].

Instead of going ‘dense’, some methods employed relatively ‘sparse’ searching approaches by introducing the concept of objectness. Lampert et al. [24] used an objectness quality function to discard sub-search spaces whose objectness scores are under a certain threshold, where object detector becomes an objectness quality function. Instead of using the object detector, Alexe et al. [1] introduces a generic objectness measure, to estimate how likely it is for a region to contain object of any category using saliency, color contrast, edge density, and boundary information. Several more approaches [5, 6, 18, 19, 33, 37] to generate RoIs based on objectness characteristics have been introduced afterwards. Recently, Ren et al. [28] introduced a region proposal network (RPN) incorporated into the CNN which also generates RoIs based on the objectness.

To garner the advantages from both of the searching approaches, ME R-CNN utilizes multi-scale sliding window (exhaustive search) along with the objectness-based (selective search) RoI generators.

## 3. The Proposed Approach

### 3.1. Architecture

As ME R-CNN shares the structural backbone of the Fast R-CNN (FRCN) [13] architecture, we briefly introduce how FRCN works to help the readers better understand the proposed architecture. FRCN consists of the per-image convolutional network and the per-RoI network. The per-image convolutional network takes an input image and computes the convolutional per-image feature map which is the output of the last convolutional layer. Meanwhile, a sparse set of RoIs is generated by the selective search. For each RoI, the per-RoI network generates a per-RoI feature map by cropping the corresponding RoI from the per-image feature map. This is then max pooled to have a fixed size output. The output size is set to match the input size of the first fully-connected layer of the predefined CNN (e.g.,  $7 \times 7$  for VGG16 [31]). The per-RoI feature map is then fed into a single stream of fully connected layers which is followed by two sibling fully connected layers. Two sibling layers are for object classification and bounding box regression.

In ME R-CNN, we remodel the two major modules of FRCN to enhance the overall detection accuracy: RoI generation module and the per-RoI network module. In terms of RoI generation, ME R-CNN acquires a combined set of RoIs generated by both the selective and exhaustive search. Instead of using a single stream per-RoI network, it adopts per-RoI multi-expert network which consists of

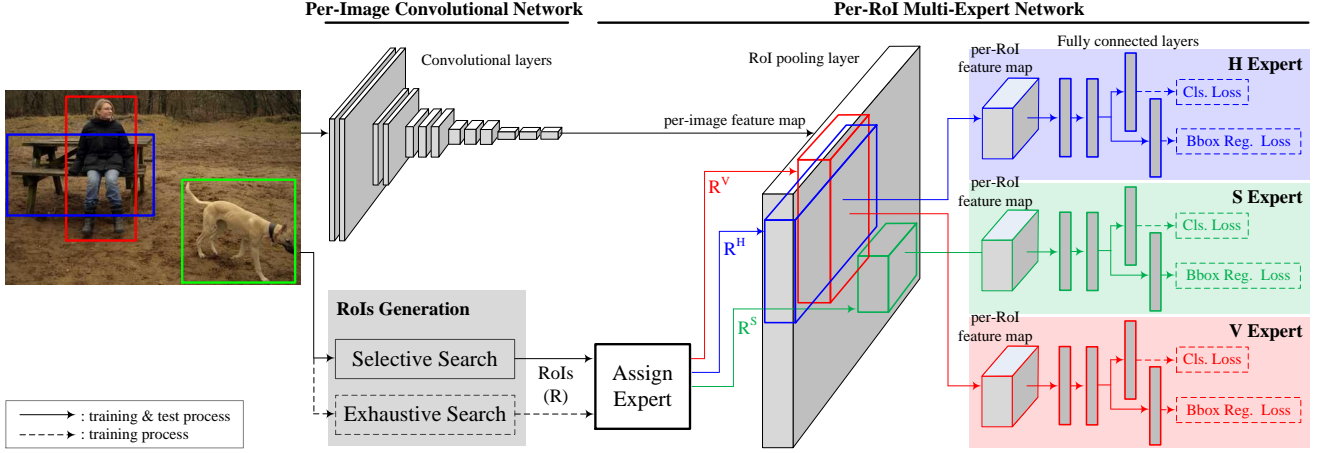


Figure 2: **ME R-CNN architecture.** Per-image convolutional network computes per-image feature map while the selective and the exhaustive search generates the RoIs (R). Then, an appropriate expert is assigned to each R. ( $R^H$ ,  $R^S$ , and  $R^V$  for H, S, and V expert, respectively.) The associated per-RoI feature maps which are the outputs of the RoI pooling layer are fed into the assigned experts. Exhaustive search is only used for the training process.

three streams each of which is called an ‘expert.’ Each expert has the same form of the fully connected layers of FRCN. Appropriate expert assignment is carried out by matching the shape of the given RoI to one of the predefined distinctive shape categories: horizontally elongated, square-like, or vertically elongated. For each RoI, its associated per-RoI feature map, which is the output of the RoI pooling layer, is fed into the assigned expert. Figure 2 illustrates the proposed ME R-CNN structure.

**RoI Augmentation.** As ME R-CNN contains larger number of parameters compared to FRCN, more examples are required to train the network. Therefore, along with the relatively sparse set of RoIs generated by the selective search [33], the proposed network also intakes a dense set of RoIs produced by multi-scale sliding windows in an exhaustive manner.

The exhaustive search looks for regions with various aspect ratios. We have used the height and width ratios of [4:1, 2:1, 1:1, 1:2, 1:4] in our experiments which are intended to cover square-like objects as well as elongated objects. For a particular aspect ratio  $r$ , the multi-scale search begins with the initial window size of width ( $w$ ) and height ( $h$ ), such that  $w/h = r$  and  $\max(w/W, h/H) = 1$ , where  $W$  and  $H$  are the width and the height of the input image, respectively. The stride is set as  $0.25 \times \min(w, h)$ . After sliding the window with a particular scale over the entire image, window size is divided by  $2^{(1/4)}$  and the window is slid again. This is to decrease the size of the window by half for every four iterations. The process is iterated until  $\min(h, w)$  is less than 25 pixels.

**Expert Assignment.** After RoIs are generated, each RoI is fed into one of the three experts according to its shape category. Each RoI is labeled with a shape category chosen among horizontally elongated (**H**), square-like (**S**), or vertically elongated (**V**) according to its aspect ratio as:

$$\text{RoI's shape category} = \begin{cases} \mathbf{H} & \text{if } \log_2(w/h) > 0.5 \\ \mathbf{V} & \text{if } \log_2(w/h) < -0.5 \\ \mathbf{S} & \text{otherwise,} \end{cases}$$

where  $h$  and  $w$  are the height and the width of an RoI, respectively. For instance, according to this rule, all RoIs whose aspect ratios are closer to 2:1 than 1:1 or 1:2, are assigned to **H** expert, where the ratios indicate  $w : h$ .

### 3.2. Learning the Network

The network whose weights are denoted as  $W$  is optimized by minimizing the loss function  $L(W)$  which is a sum of a regularization function  $R(W)$  and three pairs of loss functions, each pair being connected to one of the experts in the network. For each expert  $e$ , a softmax loss  $L_{softmax}$  and L1 smooth loss  $L_{smooth}$  are used for object classification and bounding box regression, respectively. (Details of two loss functions are described in [13].) The loss function is formularized as follows:

$$L(W) = R(W) + \sum_{e \in \{\mathbf{V}, \mathbf{H}, \mathbf{S}\}} L_{softmax}^{(e)}(W) + L_{smooth}^{(e)}(W). \quad (1)$$

As in FRCN, our network is trained using stochastic gradient descent (SGD) with three batches of 128, momentum of 0.9 and weight decay of 0.0005.



**Multi-batch Preparation.** Three batches are prepared for every iteration to optimize the three experts. Each batch is built from two images, and each image contributes 64 randomly chosen RoIs. For each expert, only the RoIs that match its associated shape category are selected for training. Each RoI is labeled as a positive or negative example according to an intersection over union (IoU) overlap criteria between the RoI and the groundtruth bounding box. The RoIs having IoU overlap bigger than 0.5 are labeled as positive examples and the ones with IoU between 0.1 and 0.5 are labeled as negative. All the remaining RoIs are excluded from training.

For each batch, the ratio between the number of positive and negative examples is fixed as 1:3. Specifically, one batch gets 16 positive and 48 negative examples from one image based on our training example balancing protocol. However, some images may not be able to provide the required number of positive and negative training examples to construct a batch which satisfies our protocol. For instance, images mostly containing horizontally elongated objects (e.g., *train*) may not have sufficient number of RoIs that are vertically elongated. To deal with such cases, we employ a strategy to take care of the exceptional cases. If the total number of positive examples is less than 16, negative examples are used instead to fill the missing amount. If the total number of positive and negative examples are less than 64, randomly chosen examples are used to fill the missing portion. We force that these randomly chosen examples have a particular aspect ratio of 2:1 (**H**), 1:1 (**S**), or 1:2 (**V**) according to the shape category of the batch.

**Finetuning.** The proposed network is finetuned from the image classification CNN pretrained over a large scale ImageNet dataset [9]. Several previous literatures [14] show that, for object detection, finetuning from such pretrained network performs significantly better than learning from scratch in terms of detection accuracy. Generally, all layers of the pretrained network, except the last fully connected layer, are used to set the initial weights of the new network. In terms of learning rate, we used a smaller value than that used to train the pretrained network. Accordingly, we use 0.001 as a base learning rate, whereas 0.01 is used to train the pretrained network.

FRCN does not finetune the first two convolutional layers because those layers tend to capture general image characteristics while other layers are more correlated with the training purpose, which is to perform the object detection. The single stream of fully connected layers of the pretrained CNN is used to finetune all three streams of fully connected layers in ME R-CNN. When finetuning the shared convolutional layers, we multiply 1/3 to the base learning rate because optimizing these layers are affected by all three streams for each training iteration when back-propagation

takes place.

The two sibling fully connected layers at the end of each expert cannot be finetuned from the pretrained image classification network as the target categories are different. To train the fully connected layer linked to the classification loss, the weights are initialized by randomly selecting them according to Gaussian distribution with the mean of 0 and the standard deviation 0.01. For the other fully connected layer linked to the bounding box regression loss, we initialized the weights randomly selected from Gaussian distribution with the mean and the standard deviation of 0 and 0.001, respectively.

### 3.3. Object Detection

For testing, we only use the selective search to generate the RoIs. ME R-CNN outputs three sets of detection results, bounding boxes and their scores, from three different experts. The bounding boxes are refined by incorporating the output of bounding box regression layers. We combine these three sets of detection results and apply non-maximum suppression (NMS) with overlap criteria of 0.3 for each object category.

## 4. Analyses of ME R-CNN

In this section, we present experimental results which demonstrate the effectiveness of adopting the multi-experts into the architecture and the RoI augmentation using the exhaustive search.

### 4.1. Experimental Setup

All the experiments shown in this section are conducted on PASCAL VOC07 [10]. According to VOC’s general protocol for object detection, trainval and test sets are used for training and testing the network, respectively. As the pretrained image classification CNN which we finetune from, we use VGG16 [31]. This is because FRCN, which is mainly compared to in this section, provides the best accuracy when it is finetuned from the VGG16. We train all methods with 80k iterations. A base learning rate is set as 0.001 and dropped to 0.0001 after 60k iterations.

### 4.2. Multiple Experts

**Multiple Experts vs. Single Expert.** As Table 1 shows, ME R-CNN outperforms the single expert network (FRCN w/ Augmented RoIs) by 0.9%. Note that these methods use the RoIs generated by both the exhaustive and the selective search. Even without the augmented set of RoIs, ME R-CNN outperforms FRCN by 1.4%.

**Layer Sharing in Multi-Experts.** Multi-experts in ME R-CNN contain three times more number of fully connected layers compared to FRCN, which brings up the memory efficiency issues. Table 2 shows the detection performances

Table 1: Effects of two major modules of ME R-CNN (SS and ES are the selective search and the exhaustive search, respectively.)

Method	RoIs	No. of Experts	mAP (%)
FRCN [13]	SS	1	66.9
FRCN w/ Augmented RoIs	SS+ES	1	68.1
ME R-CNN w/o Augmented RoIs	SS	3	68.3
ME R-CNN	SS+ES	3	<b>69.0</b>

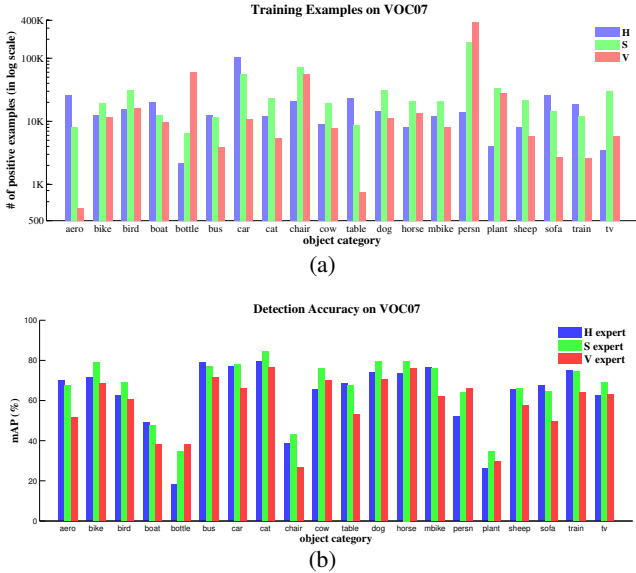


Figure 3: **Comparison of Multiple Experts.** (a) The number of training examples to be used for training three experts. (b) Detection accuracy achieved by three experts.

acquired by employing three different structural variations on which fully connected layers are shared across the multiple experts. We observe that sharing fc6 layer across the experts is a reasonable compromise between the memory efficiency and the detection performance. Similar to optimizing the convolutional layers, shared fully connected layers are optimized by multiplying 1/3 to the base learning rate.

Table 2: Effects of layer sharing across multi-experts

	FRCN	layers shared in ME R-CNN		
		none	fc6	fc6 & fc7
mAP (%)	66.9	68.9	<b>69.0</b>	68.6

**Comparison among Three Experts.** Table 3 shows the detection performance achieved by separate experts. This ex-

periment is carried out to observe the general performance of each expert over the overall set of RoIs instead of evaluating each expert with only the RoIs in the corresponding shape category. The **S** expert (67.1%) performs better than the **H** and **V** experts, which is comparable to the performance obtained by FRCN with augmented RoIs (68.1%). The **S** expert not only covers the square-like objects, but it seems to have the ability to detect the object instances that fall into the other shapes (**H** or **V**). When these three experts are used in a unified network setting of ME R-CNN, the performance is boosted up to 69.0% (Table 1).

Table 3: Detection accuracy of multiple experts

Expert	<b>H</b>	<b>S</b>	<b>V</b>
mAP (%)	61.4	67.1	58.7

Figure 3a shows the distribution of training examples of different shape categories (i.e., **H**, **S**, and **V**) for each object category. Figure 3b depicts the detection accuracies achieved by different experts for each object category. For most object categories, the accuracies for **H**, **S**, and **V** follow the distribution trend shown in 3a. That is, when an object category mostly contains examples with certain shape, the best detection accuracy was obtained by the expert responsible for that shape. For instance, **H** expert performs the best in detecting horizontally elongated objects such as *aeroplane*, *car*, and *sofa* while **V** expert shows the best performance in *bottle* and *person* object categories which mostly show vertically elongated shapes.

### 4.3. RoI Augmentation

**Effects of RoI Augmentation.** In Table 1, we show that the performance of the original FRCN can be increased by 1.2% just by using the combined RoIs (the exhaustive and selective search) instead of using the RoIs generated by the selective search only. Using the combined RoIs for ME R-CNN also brings a boosted performance by 0.7% than the case when only the selective search is used.

**Exhaustive Search vs. Selective Search.** The exhaustive search has two complementary roles compared to the selective search: (i) it provides RoIs which capture objects missed by the selective search, and (ii) it can provide more positive and negative training examples.

Figure 4a shows the recall achieved by the RoIs whose IoU overlaps with the groundtruth are over varying thresholds on PASCAL VOC07 dataset [10]. The graph indicates that when we set the IoU threshold conservatively, the selective search provides better recall over the exhaustive search. On the other hand, when considering the positive example selection criteria (i.e., at 0.5 IoU threshold), the exhaustive search achieves better recall by 7.4% than the selective

Table 4: **VOC 2007 test** detection average precision. All methods use VGG16. ME R-CNN inherits the FRCN architecture. Training set key: **07**: VOC07 trainval, **07+12**: union of VOC07 trainval and VOC12 trainval.

method	train set	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	persn	plant	sheep	sofa	train	tv
FRCN [13]	07	66.9	74.5	78.3	69.2	53.2	36.6	77.3	78.2	82.0	40.7	72.7	67.9	79.6	79.2	73.0	69.0	30.1	65.4	70.2	75.8	65.8
ME R-CNN	07	<b>69.0</b>	70.6	<b>78.9</b>	68.2	55.8	44.3	80.9	78.2	84.6	44.4	76.5	70.4	80.6	81.5	76.6	70.8	<b>35.1</b>	66.4	69.9	76.8	69.5
FRCN [13]	07+12	70.0	77.0	78.1	69.3	59.4	38.3	81.6	78.6	86.7	42.8	78.8	68.9	84.7	82.0	76.6	69.9	31.8	70.1	74.8	80.4	70.4
ME R-CNN	07+12	<b>72.2</b>	<b>78.1</b>	<b>78.9</b>	<b>69.4</b>	<b>61.3</b>	<b>44.5</b>	<b>84.8</b>	<b>81.7</b>	<b>87.6</b>	<b>50.7</b>	<b>80.1</b>	<b>70.6</b>	<b>85.8</b>	<b>84.8</b>	<b>78.7</b>	<b>72.3</b>	35.0	<b>71.9</b>	<b>75.3</b>	<b>79.9</b>	<b>72.7</b>

Table 5: **VOC 2007 test** detection average precision. All methods use ResNet-101. ME R-CNN inherits the Faster R-CNN architecture.

method	train set	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	persn	plant	sheep	sofa	train	tv
Faster R-CNN [17]	07+12	76.4	79.8	80.7	76.2	68.3	55.9	85.1	85.3	89.8	56.7	87.8	69.4	88.3	88.9	80.9	78.4	41.7	78.6	79.8	<b>85.3</b>	72.0
ME R-CNN	07+12	<b>78.7</b>	<b>81.2</b>	<b>81.9</b>	<b>78.0</b>	<b>71.8</b>	<b>65.0</b>	<b>86.0</b>	<b>87.5</b>	<b>91.3</b>	<b>61.0</b>	<b>89.2</b>	<b>69.9</b>	<b>88.4</b>	<b>90.1</b>	<b>83.9</b>	<b>81.4</b>	<b>45.2</b>	<b>81.0</b>	<b>81.7</b>	<b>85.3</b>	<b>73.9</b>

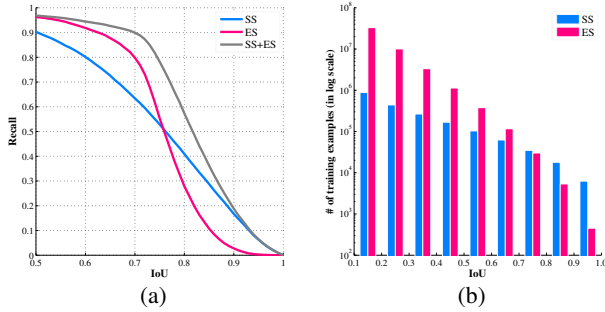


Figure 4: **Exhaustive search vs. selective search.** (a) Recall achieved by RoIs whose IoU overlaps with groundtruth are over varying thresholds. (b) The histogram of RoIs w.r.t. their IoU overlaps with groundtruth.

search. This supports the first complementary role of the exhaustive search. Based on this observation, we use the combined set of RoIs generated by both searching strategies.

Figure 4b show a histogram of RoIs with respect to their IoU overlaps with the groundtruth. In the entire training examples, the exhaustive search generates significantly larger number of RoIs than the selective search. This verifies the second role of the exhaustive search.

## 5. Evaluation on PASCAL VOC 07 and 12

### 5.1. Experimental Setup

We use either VGG16 [31] or ResNet-101 [17] as a pre-defined CNN for all experiments. Like the previous section, VGG16 is inherited in the similar architecture of FRCN. Meanwhile, for ResNet-101, we use the architecture of Faster R-CNN rather than that of FRCN because combina-

tion of ResNet-101 and Faster R-CNN shows the best detection accuracy in PASCAL VOC12 competition.

In a case that we use VGG16, the network is learned by the end-to-end optimization. On the PASCAL VOC 07 trainval set, training ME R-CNN follows the training schedule described in the section 4.1. For the PASCAL VOC 12 trainval set, we use SGD optimization with a base learning rate of 0.001, 120k iterations and decay step size of 90k iterations. When using extra dataset such as **07+12** (Union of VOC 07 trainval and 12 trainval) and **07++12** (Union of VOC 07 trainval, VOC 07 test, and VOC 12 trainval), we use 200k iterations with decay step size of 150k iterations and 240k iterations with decay step size of 180k, respectively.

When using ResNet-101 and Faster R-CNN architecture, we use four-step alternating optimization which iteratively optimizes either RPN or the remaining layers in turn following the original training procedure [17, 28]. For RPN training which is performed in the first and third optimization step, we use 200k iterations with decay step size of 150k. For training the remaining portion of the network in the second and fourth steps, we use 200k iterations with decay step size of 150k for **07+12** and 240k iterations with decay step size of 180k for **07++12**. For all evaluations in Section 5, we use single-scale training/testing as in [13], by setting the shorter side of the images to be 600 pixels.

We have carried out all the experiments on Caffe framework [20] with a Titan X GPU.

**Adopting Multi-Expert into ResNet-101.** For object detection, He et al. [17] assigns the last 10 convolutional layers of ResNet-101 to function as the per-RoI network. We only use the last 6 convolutional layers as the per-RoI multi-expert network and insert the first 4 convolutional layers

Table 6: **VOC 2012 test** detection average precision. All methods use VGG16. ME R-CNN inherits the FRCN architecture. Training set key: **12**: VOC12 trainval, **07++12**: union of VOC07 trainval, VOC07 test, and VOC12 trainval.

method	train set	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	persn	plant	sheep	sofa	train	tv
FRCN [13]	12	65.7	80.3	74.7	66.9	46.9	37.7	73.9	68.6	87.7	41.7	71.1	51.1	86.0	77.8	79.8	69.8	32.1	65.5	63.8	76.4	61.7
ME R-CNN <sup>1</sup>	12	<b>67.8</b>	82.6	76.4	69.9	50.3	41.8	75.5	71.1	87.0	42.0	74.3	56.0	86.3	81.5	78.9	72.4	34.1	68.5	62.6	79.6	64.7
FRCN [13]	07++12	68.4	82.3	78.4	70.8	52.3	38.7	77.8	71.6	<b>89.3</b>	44.2	73.0	55.0	87.5	80.5	80.8	72.0	35.1	68.3	65.7	80.4	64.2
ME R-CNN <sup>2</sup>	07++12	<b>70.7</b>	<b>84.0</b>	<b>79.8</b>	<b>72.4</b>	<b>54.9</b>	<b>43.3</b>	<b>78.4</b>	<b>74.7</b>	<b>89.3</b>	<b>46.6</b>	<b>76.1</b>	<b>60.6</b>	<b>87.8</b>	<b>83.6</b>	<b>82.1</b>	<b>74.8</b>	<b>39.4</b>	<b>70.6</b>	<b>65.7</b>	<b>82.5</b>	<b>67.9</b>

<sup>1</sup><http://host.robots.ox.ac.uk:8080/anonymouse/69D0YS.html>

<sup>2</sup><http://host.robots.ox.ac.uk:8080/anonymouse/PLPKPU.html>

Table 7: **VOC 2012 test** detection average precision. All methods use ResNet-101. ME R-CNN inherits the Faster R-CNN architecture.

method	train set	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	persn	plant	sheep	sofa	train	tv
Faster R-CNN [17]	07++12	73.8	86.5	81.6	<b>77.2</b>	58.0	51.0	78.6	76.6	<b>93.2</b>	48.6	<b>80.4</b>	59.0	<b>92.1</b>	<b>85.3</b>	84.8	80.7	48.1	77.3	66.5	<b>84.7</b>	65.6
ME R-CNN <sup>3</sup>	07++12	<b>76.1</b>	<b>87.1</b>	<b>82.7</b>	76.3	<b>62.5</b>	<b>62.6</b>	<b>81.7</b>	<b>80.8</b>	90.6	<b>54.8</b>	79.1	<b>63.1</b>	89.6	84.4	<b>85.4</b>	<b>84.1</b>	<b>55.0</b>	<b>77.9</b>	<b>67.1</b>	84.3	<b>71.9</b>

<sup>3</sup><http://host.robots.ox.ac.uk:8080/anonymouse/M9ZUJK.html>

into the per-image convolutional network due to the GPU memory limitation. We also reduce the batch size from 128 to 64 for training.

## 5.2. PASCAL VOC 07 and 12 Results

Table 4 shows that, on VOC07, ME R-CNN provides improved detection accuracy in mAP than FRCN when using VOC07 trainval set for training (69.0% vs. 66.0%). When using **07++12**, ME R-CNN outperforms FRCN by similar amount of increase (72.2% vs. 70.0%). VOC12 results are shown in Table 6 where we observe consistent performance boost for ME R-CNN. In both cases of VOC12 trainval (67.8% vs. 65.7%) and **07++12** (70.7% vs. 68.4%), ME R-CNN outperforms FRCN by more than 2% mAP.

ME R-CNN shows a consistent performance boost when compared with ResNet-101 with Faster R-CNN on both VOC07 (78.7% vs. 76.4%) and VOC12 (76.1% vs. 73.8%). For this result, ME R-CNN was built on top of the ResNet-101 with Faster R-CNN architecture for fair comparison, also showing that the proposed architecture can effectively be combined with various types of object detection CNN architectures.

## 6. Future Works

In this paper, we have focused on showing that ME R-CNN architecture can boost the performance of when integrated with the baseline object detection CNNs. Fast R-CNN and Faster R-CNN, which are two of the most widely used object detection networks, were selected to demonstrate the effectiveness.

In the future, we will focus on producing the state-of-the-art performance in PASCAL VOC datasets by incorporating additional processings (referred to as 'adding bells and whistles' in [30]) such as multi-scale training/testing [13, 16],

online hard example mining (OHEM) [30], iterative bounding box regression [12], global context [17], ensemble of classifiers [17], and integrating with image classification output [4]. We will evaluate ME R-CNN in large-scale Microsoft COCO dataset [25] as well. We also plan to incorporate multi-expert into other CNN-based detection architecture such as SSD [26] and R-FCN [7] which has recently been introduced.

## 7. Conclusion

We introduced ME R-CNN which uses multiple experts in place of a conventional single classifier incorporated in CNN-based object detection architectures. Compared to the single model, multiple experts is known to better capture variations in basic shape categories as well as object appearance caused by different poses and viewing angles. We categorized given regions-of-interest (RoIs) into three pre-defined distinctive shape categories: horizontally elongated, square-like, and vertically elongated. Then an appropriate expert is assigned to each RoI according to the shape category of the RoI.

To provide an augmented set of RoIs, we use two methods: the selective search and the exhaustive search. While the selective search produces a sparse set of RoIs, which results in reducing computational complexity for object detection, the exhaustive search provides two complementary roles: 1) the exhaustive search is able to search regions missed by the selective search and 2) provides an incomparably large number of RoIs. For testing, we only use a sparse set of RoIs generated by the selective search to maintain the efficiency of the testing process. With benefits of these two major modules, ME R-CNN proves its effectiveness in enhancing the detection accuracy in PASCAL VOC 07 and 12 dataset over the baseline methods.



## References

- [1] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *CVPR*, 2010. 3
- [2] P. Arbelvez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *CVPR*, 2014. 3
- [3] E. Bernstein and Y. Amit. Part-based statistical models for object classification and detection. In *CVPR*, 2015. 3
- [4] Y. Cao\*, H. Lee\*, and H. Kwon. Enhanced object detection via fusion with prior beliefs from image classification. *arXiv preprint arXiv:1610.06907*, 2016. 8
- [5] J. Carreira and C. Sminchisescu. CPMC: Automatic object segmentation using constrained parametric min-cuts. *TPAMI*, 2012. 3
- [6] N. Chavali, H. Agrawal, A. Mahendru, and D. Batra. Object-proposal evaluation protocol is ‘gameable’. In *CVPR*, 2016. 3
- [7] J. Dai, Y. Li, K. He, and J. Sun. R-FCN: Object detection via region-based fully convolutional networks. In *NIPS*, 2016. 1, 2, 3, 8
- [8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 3
- [9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 1, 5
- [10] M. Everingham, L. Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) challenge. *IJCV*, 2015. 5, 6
- [11] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 2010. 1, 3
- [12] S. Gidaris and N. Komodakis. Object detection via a multi-region & semantic segmentation-aware cnn model. In *ICCV*, 2015. 8
- [13] R. Girshick. Fast R-CNN. In *ICCV*, 2015. 1, 2, 3, 4, 6, 7, 8
- [14] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 1, 2, 3, 5
- [15] H. Harzallah, F. Jurie, and C. Schmid. Combining efficient object localization and image classification. In *CVPR*, 2009. 3
- [16] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, 2014. 1, 2, 3, 8
- [17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 7, 8
- [18] J. Hosang, R. Benenson, P. Dollar, and B. Schiele. What makes for effective detection proposals? *TPAMI*, 2016. 3
- [19] J. Hosang, R. Benenson, and B. Schiele. How good are detection proposals, really? In *BMVC*, 2014. 3
- [20] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACMMM*, 2014. 7
- [21] F. Khan, R. Anwer, J. van de Weijer, A. Bagdanov, M. Vanrell, and A. Lopez. Color attributes for object detection. In *CVPR*, 2012. 1, 3
- [22] I. Kokkinos. Rapid deformable object detection using dual-tree branch-and-bound. In *NIPS*, 2011. 3
- [23] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1, 2
- [24] C. Lampert, M. Blaschko, and T. Hofmann. Efficient sub-window search: A branch and bound framework for object localization. *TPAMI*, 2009. 3
- [25] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 8
- [26] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. In *ECCV*, 2016. 8
- [27] T. Malisiewicz, A. Gupta, and A. Efros. Ensemble of exemplar-SVMs for object detection and beyond. In *ICCV*, 2011. 1, 3
- [28] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 1, 2, 3, 7
- [29] H. Schneiderman and T. Kanade. A statistical method for 3D object detection applied to faces and cars. In *CVPR*, 2000. 3
- [30] A. Shrivastava, A. Gupta, and R. Girshick. Training region-based object detectors with online hard example mining. In *CVPR*, 2016. 3, 8
- [31] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 1, 3, 5, 7
- [32] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 1, 2
- [33] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. *IJCV*, 2013. 3, 4
- [34] S. Vijayanarasimhan and K. Grauman. Efficient region search for object detection. In *CVPR*, 2011. 3
- [35] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001. 3
- [36] K. Yi, K. Yun, S. Kim, H. Chang, and J. Choi. Detection of moving objects with non-stationary cameras in 5.8ms: Bringing motion detection to your mobile device. In *CVPR Workshop*, 2013. 3
- [37] L. Zitnick and P. Dollar. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014. 3